Pedestrian Tracking Using Offboard Cameras

Olivier Aycard, Anne Spalanzani, Julien Burlet, Chiara Fulgenzi, Trung-Dung Vu, David Raulo, Manuel Yguel

Pedestrians Tracking Using Offboard Cameras

Olivier Aycard, Anne Spalanzani, Julien Burlet, Chiara Fulgenzi, Trung Dung Vu, David Raulo, Manuel Yguel
GRAVIR-IMAG & INRIA RA
655 avenue de l'Europe - Montbonnot
38334 Saint Ismier Cedex - FRANCE
Email : FirstName.LastName@imag.fr

***Abstract*— In this paper, we detail the hardware and software perception system designed and developed to track pedestrians using a set of offboard cameras. It has been used in the context of vulnerable safety in a car park. This architecture is divided in two parts: a fusion part to fusion the data given by the set of offboard cameras and a tracking part to sequentially estimate the position of each pedestrian present in the environment and to determine the number of pedestrians. Finally, some experimental results are presented.**

Keywords: perception, sensor data fusion, pedestrians tracking

I. Introduction

In France, about 33% of roads victims are VRU[1]. In its 3^{rd} framework, the french PREDIT[2] includes VRU Safety. The PUVAME project [1] was created to generate solutions to avoid collisions between VRU and Bus in urban traffic. An accident analysis has shown that an important part of these collisions take place at intersection and bus stop. To reduce accidents, a first requirement of the PUVAME project is to design and develop a perception system in order to track pedestrians present at these particular places. This objective is achieved using a combination of offboard cameras, observing intersections or bus stops, to detect VRU present at intersection or bus stop.

Typical perception systems report measurements from diverse sensors, such as radar, laser or camera. Sensor data fusion is a prerequisite to exploit the inherent advantages of multi sensors perception systems over single sensor systems. Objects tracking is also an integral part of perception systems employing one or more sensors to interpret the environment. Unfortunately, the multi objects tracking problem is complex. Firstly, measurements have to be assigned to an object to reestimate its position. These assignments are generally unknown. Moreover objects may be occluded and some measurements correspond to any objects (ie, false alarms). As long as the association is considered in a deterministic way, the possible associations must be exhaustively enumerated. This leads to an NP-hard problem because the number of possible associations increases exponentially with the number of sensors and objects. Several association methods have been presented in [9]. Finally, one also has to solve the problem of estimating the number of objects that are currently in the field of view.

[1]Vulnerable Road Users

[2]Programme de Recherche et d'Innovation dans les Transports Terrestres

In many automotive applications [2] [7], the association of sensor data to objects is done using a gating approach. In this approach, the uncertainty associated to the actual position of a vehicle is modelized by a Gaussian. When one observation has an important probability (ie, superior than a given threshold) to correspond to this gaussian, it is associated to the corresponding object. Finally, the observations corresponding to the same object are used to reestimate its position. This solution has the main advantage to be simple, fast and to drastically decrease the number of possible association. However, it only works in non cluttered environment.

In our perception solution, we propose a different approach to perform association. Before performing association, we perform fusion of data given by different sensors to build a map of the current environment (ie, a snapshoot of the current environment). In a second step, using this map, we search the pedestrians currently present in the environment. Finally, we associate this list of pedestrians with the list of pedestrians previously present in the environment. As the number of pedestrians currently present in the environment is almost always inferior to the number of observations, the number of possible associations decreases.

To model the environment and to perform multi-sensor fusion, we use a generic framework called Occupancy Grids (OG). This framework has been introduced by Elfes and Moravec at the end of the 1980s. An occupancy grid is a stochastic tesselated representation of spatial information that maintains probabilistic estimates of the occupancy state of each cell in a lattice [4]. The main advantage of this approach is the ability to integrate several sensors in the same framework taking the inherent uncertainty of each sensor reading into account, in opposite to the Geometric Paradigm [3] whose method is to categorize the world features into a set of geometric primitives. The alternative that OGs offer is a regular sampling of the space occupancy, that is a very generic system of space representation when no knowledge about the shapes of the environment is available. And all the more so as with appropriate sensor models OG provide a rigorous way to manage occlusions in the sensor field of view. On the contrary of a feature based environment model, the only requirement for an OG building is a bayesian sensor model for each cell of the grid and each sensor. This sensor model is the description of the probabilistic relation that links sensor measurement to space state, that OG necessitates to make the sensor integration.

In this paper, we detail the solution we develop to track pedestrians using information about their position given by a set of offboard cameras. This solution is divided in 2 main parts:

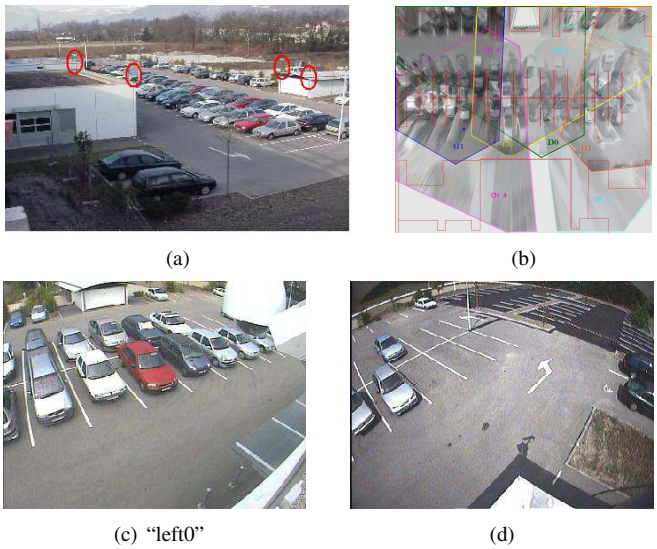
- A fusion part that allows to use data coming from different sensors in order to compute a better estimation of the position of each pedestrian [10]. It also increases the field of view of the whole perception system, and is useful to decrease the level of false alarms. This fusion part firstly builds an occupancy grid using data coming from a set of offboard cameras [12]. In a second step, pedestrians are extracted from this grid;
- A tracking part that associates pedestrians currently present in the environment with pedestrians previously present in the environment and estimates the number of pedestrians present in the environment and the position of each pedestrian.

In next section, we present the experimental platform used to evaluate the solution we propose. Section III details the two level architecture used to track pedestrians. Experimental results are reported in section IV. We give some conclusions and perspectives in section V.

II. ParkNav platform

The experimental setup used to evaluate our fusion scheme is an evolution of the ParkView platform, initially developped for a French national project designed for the Interpretation of Complex Dynamic Scenes and Reactive Motion Planning.

The ParkView platform is composed of a set of six off-board analog cameras, installed in a car-park setup such as their field-of-view partially overlap (see figure 1), and three Linux(tm) workstations in charge of the data processing, connected by a standard Local Area Network.



(a) (b) (c) “left0” (d)

Fig. 1. (a) Location of the cameras on the parking; (b) Field-of-view of the cameras projected on the ground; (c) View from one camera

The workstations run a specifically developped client-server software composed of three main parts, called the *map server*, the *map clients* and the *connectors* (figure 2).

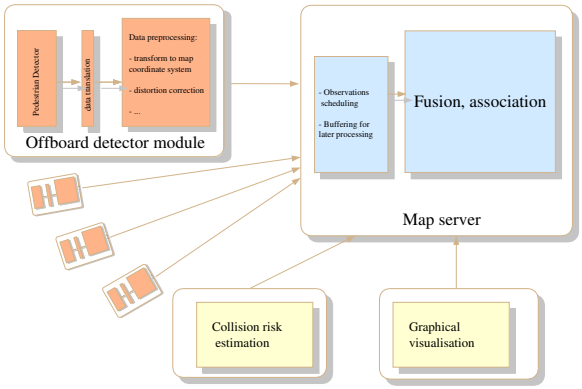


Fig. 2. The ParkView platform software organization

The *map server* processes all the incoming observations provided by the different clients, in order to maintain a global high-level representation of the environment; this is where the data fusion occurs. A single instance of the server runs.

The *map clients* connect to the server and provide the users with a graphical representation of the environment; they can also process this data further and perform application-dependant tasks. For example, in a driving assistance application, the vehicle on-board computer will be running such a client specialized in estimating the collision risk.

The *connectors* receive the raw sensor-data, perform the pre-processing, and send the resulting *observations* to the map server. Each computer connected with one or several sensors runs such a *connector*. For the application described here, all data preprocessing basically consist in detecting pedestrians. Therefore, the video stream of each camera is processed independently by a dedicated detector. The role of the detectors is to convert each incoming video frame to a set of bounding rectangles, one by target detected in the image plane. The detector observation consists in a set of rectangles detected at a given time. It is sent to the map server.

Since the fusion system operates in a fixed coordinate system, distinct from each of the camera’s local system, a coordinate transformation must be performed. For this purpose, each camera has been calibrated beforehand. The result of this calibration consists in a set of parameters:

- the intrinsic parameters contain the information about the camera optics and CCD sensor: the focal length and focal axis, the distorsion parameters,
- the extrinsic parameters consist of the homography matrix: the 3x3 homogenous matrix transforms the coordinates of an image point to the ground coordinate system.

In such a multi-sensor system, special care must be taken of proper timestamping and synchronization of the observations. This is especially true in a networked environment, where the standard TCP/IP protocol would incur its own latencies.

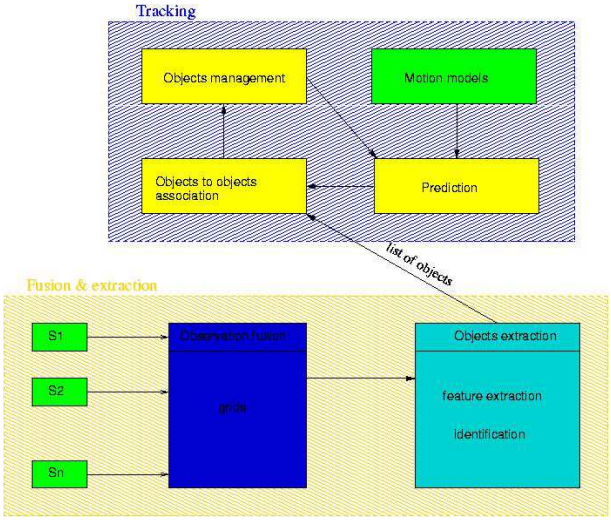


Fig. 3. Architecture of the pedestrians tracker

The ParkView platform achieves the desired effect by using a specialized transfer protocol, building on the low-latency properties of UDP while guaranteeing in-order, synchronised delivery of the sensor observations to the server.

III. Architecture of the pedestrians tracker

Our objective is to have a robust perception using multi-sensor approaches to track the different pedestrians present in the car park. The whole architecture, depicted in figure 3, is made of 2 levels : one focuses on the fusion of observations given by several sensors and the objects' extraction, the other focuses on the objects' tracking. In this section, the different modules of our architecture are described.

A. Fusion and extraction level

Observations come from a pedestrian detector, they are merged thanks to an occupancy grid and then, objects are extracted from this grid.

1) Pedestrian detector: To detect VRUs present in the car parkbackground extraction is used to detect objects present in the image. In a second phase, a pedestrian detector using learning methods [11] is used to detect pedestrians.

2) Occupancy grid: The construction of the occupancy grid as a result of the fusion of the detector observations given by different cameras is detailed in [12]. In this paragraph, we only give an overview of the construction of this occupancy grid. The observations come from sensors and the heart of the modelling problem is to define how each sensor measure modify the cell state.

a) Mathematical Framework: we introduce our framework and notation, deriving the update equations of a cell of the grid at each sensor measurement.

- $\overrightarrow{Z} = (Z_1, \ldots, Z_s)$ a vector of s random variables, one variable for each sensor. We consider that each sensor i can return measurements from a set $\mathcal{Z}_i$ plus a special event "nothing measured" which means that the entire scanned region is free.
- $E_x \in \mathcal{E} \equiv \{\text{occ}, \text{emp}\}$. E_x is the state of the bin x either occupied ("occ") or empty ("emp"), where $x \in \mathcal{X}$. $\mathcal{X}$ is the set of indexes of all the cells in the monitored area.

For a certain variable V we will note in capital case the variable, in normal case v one of its realisation, and we will note $P(v)$ for $P([V = v])$ the probability of a realisation of the variable.

b) Joint probabilistic distribution: The lattice of cells is a type of markov field and many assumptions could be made about the dependencies between cells and especially adjacent cells in the lattice [6]. In this paragraph, we will explain sensor models for independent cells i.e. without any dependencies, which is a strong hypothesis but very efficient in practice since any calculus could be made for each cell apart. It leads to the following expression of a joint distribution for each cell.

$$P(E_x, \overrightarrow{Z}) = P(E_x) \prod_{i=1}^{s} P(Z_i|E_x) \tag{1}$$

Given a vector of sensor measurements $\overrightarrow{z} = (z_1, \ldots, z_s)$ we apply the bayes rule to derive the probability of cell x to be occupied:

$$P(e_x|\overrightarrow{z}) = \frac{P(e_x) \prod_{i=1}^{s} P(z_i|e_x)}{P(\text{occ}) \prod_{i=1}^{s} P(z_i|\text{occ}) + P(\text{emp}) \prod_{i=1}^{s} P(z_i|\text{emp})} \tag{2}$$

For each sensor i, the two conditional distributions $P(Z_i|\text{occ})$ and $P(Z_i|\text{emp})$ must be specified. That what is called *the sensor model* definition.

c) Building a sensor model: The problem is that motion detectors give information in the image space and that we search to have knowledge in the ground plan. We solve this problem projecting the bounding box in the ground plan using some hypothesis: we mainly suppose that the ground is a plan, all the VRU stand on the ground and the complete VRUs is visible for the camera. To build the sensor model, we first search to segment the ground plan in three types of region: occupied, occulted and free zones using the bounding boxes informations. Then we introduce an uncertainty management, using a gaussian convolution, to deal with the position errors in the detector. Finally, we convert this information into probability distributions. In the 3 next paragraphs, we detail the different steps of the construction of the sensor model.

1) Image of the ground occupation
   a) One video camera, one bounding box The inputs of this environment modelling are output of video camera detectors that give bounding boxes of detected moving objects. A video camera only sees the visible surface of the objects in its field of view. Thus we have to draw on the ground the occupied, occluded and free zones. First, we calculate the projection of the bounding box to the ground and we mark this area as occluded.

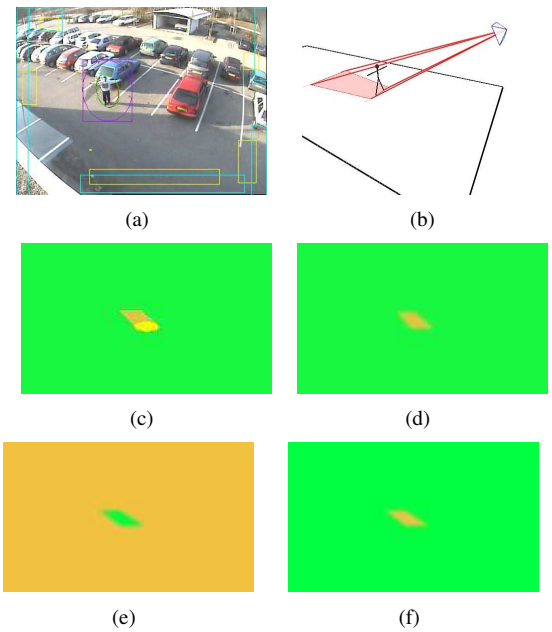


Fig. 4. (a) An image of a moving object acquired by one of the offboard video cameras and the associated bounding boxe found by the detector. (b) The occulted zone as the intersection of the viewing cone associated with the bounding boxe and the ground plan. (c) The associated ground image produce by the system. (d) Ground image after gaussian convolution with a support size of 7 pixels. (e) Probability of the ground image pixel value, knowing that the pixel corresponds to an empty cell: $P(Z|\text{emp})$ for each cell. (f) Probability of the ground image pixel value, knowing that the pixel corresponds to an occupied cell: $P(Z|\text{occ})$ for each cell.

It is possible because, the extrinsec parameters of the video camera have been calibrated before and the ground plan identified. Thus the projection of the bounding box is just the intersection of the cone of view defined by the bounding box with the ground plan (Fig. 4(b)). Second, we search for the segment corresponding to the bottom of the projected bounding box. We draw an ellipsoid around this segment and mark it as an occupied area. We draw the rest as free (Fig. 4(c)).

b) One video camera, several bounding boxes In the case of several bounding boxes, the projection of one can overlap the projection of an other. So that we have to handle carefully the order of area drawing such as no occupied area will be marked as occluded or free when it was marked as occupied before. So we define three values: $\{0.1; 0.7; 0.9\}$ for free, occluded and occupied respectively. These three values are chosen according to the uncertainty of the pedestrian process and the semantic we want to attach to each area. First we paint all the ground in free. Then we draw each bounding box with its occluded and occupied area and for each pixel the new value is just set to the max of the precedent value and the measured value.

2) Position uncertainty To handle position uncertainty due to video camera vibration, noise in the video detector, non perfect synchronisation of all the sensor measurements or the communication latency we just make a convolution of the ground image obtained in the precedent step, with a gaussian 2D-kernel. The variances of these kernels are important parameter and in fact it suits the worst of the precedent sources of position uncertainties, Fig. 4(d).

3) Building the two maps of probabilities: $P(Z|Ex)$ For each bin the precedent step provide a floating number $z \in [0; 1]$ describing the fact that there is or not an obstacle in the cell. A possible definition of the probability of this number for each possible state of the cell: emp, occ is in term of probability density:

$$p(z|\text{emp}) = 2(1-z) \quad (3)$$
$$p(z|\text{occ}) = 2z \quad (4)$$

The main information is that the close z is to 1, the most probable is the measure z, if the cell is occupied. For $P(Z|\text{occ})$ any increasing function over $[0; 1]$ which integral is 1.0 suits. Symetrically for $P(Z|\text{emp})$ any decreasing function over $[0; 1]$ which integral is 1.0 suits. We chose very simple functions: that are linear functions and reach 0 and 1 at their maxima.

d) Results: Figure 4 illustrates the whole construction of the sensor model.

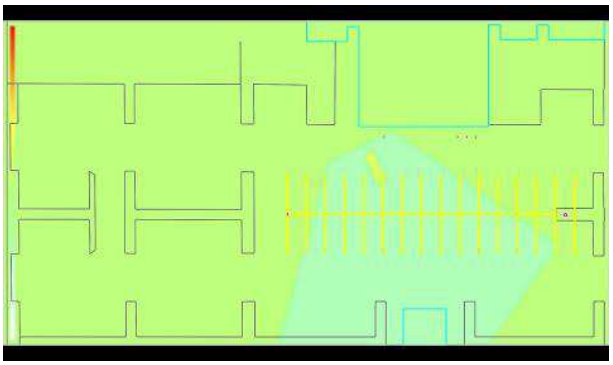


Fig. 5. The resulting probability that the cells are occupied after the inference process with one camera.

Figure 5 shows experiments (ie, the resulting occupancy grid) with one camera. One pedestrian is present in the environment. The blue area (corresponding to low probabilities of occupancy) shows the camera's field of view. The green area corresponds to the part of the environment (ie, the unknown part) that is not seen by the camera: probability equals to 0.5. The yellow area corresponds to the occluded area (ie, area behind the pedestrian).

Figure 6 shows the same pedestrian seen by two cameras. The red area corresponds to the most probable position of the pedestrian: this area is the result of the fusion of the two yellow areas given the two cameras. The 3 green areas around

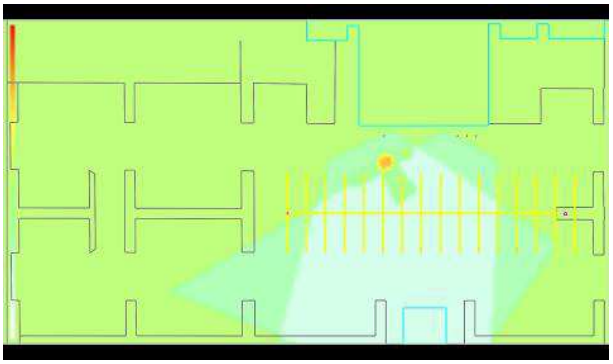


Fig. 6. The resulting probability that the cells are occupied after the inference process with two cameras.

the pedestrian correspond to the fusion between the occluded area of one camera with the free area of the other one. The area seen as free by the two cameras has a very low probability of occupancy. The 4 areas seen as free by one camera and out of the field of view of the second camera have a low probability of occupancy.

3) Object extraction: Once an occupancy grid is obtained, we want to extract the possible objects (VRUs) which are likely located in regions with high occupation probability Object-regions may have arbitrary shapes and are generally discriminant from background. From these characteristics, we apply a threshold segmentation method.

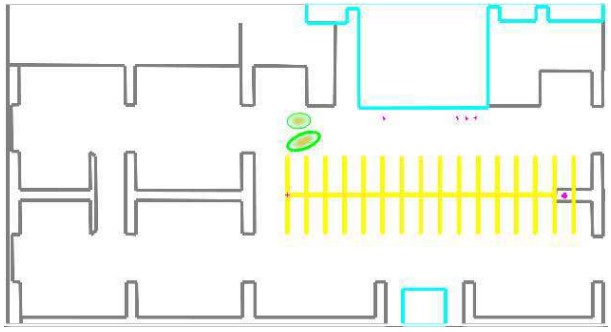


Fig. 7. detection of objects approximated with ellipses

First, an adaptive threshold is computed based on a discrete histogram of cell occupation probability values and the threshold is chosen as the mean value of the histogram. We use this threshold to transform the grid into a binary image where positive pixels represent occupied areas. In the next step a two pass segmentation algorithm is applied to extract all 4-connected groups of cells. Each connected group corresponding to a possible object is finally approximated by an ellipse represented by mean value and covariance matrix of the corresponding region (see Figure 7).

B. Tracking part

1) Prediction: Each VRU present in the environment is tracked using a Kalman filter [5]. The state vector is represented by both position and velocity of the VRU and the predicted state is computed using a constant velocity dynamical model.

2) Object to Object association: To update the position of each VRU using Kalman filter, we first need to associate the observations extracted from the occupancy grid to the predicted positions. As there could be at most one observation associated to each given VRU: a gating procedure is first applied to reduce number of possible assignments, then a global nearest neighbor data association method is used [8]. The association is also useful to manage the list of VRUs present in the environment as described in the next paragraph.

3) Object management: Each VRU is tagged with a specific ID, its position in the environment and the associated velocity. At the beginning of the process, the list of VRU present in the environment is empty. The result of the association phase is used to update this list. Several cases could appear:

1) An observation is associated to a VRU: the position and velocity of this VRU is estimated with a Kalman filter, the predicted state and this observation;
2) A VRU has no observation associated to itself: the reestimated position and velocity of this VRU are given by the predicted state;
3) An observation is not associated to any VRU: a new temporary VRU ID is created, its position is initialized at the value of the observation and its velocity is set to 0. To avoid to create VRU corresponding to false alarms, the temporary VRU is only confirmed (ie, becomes a definitive VRU) if it is seen during 3 consecutive instants.

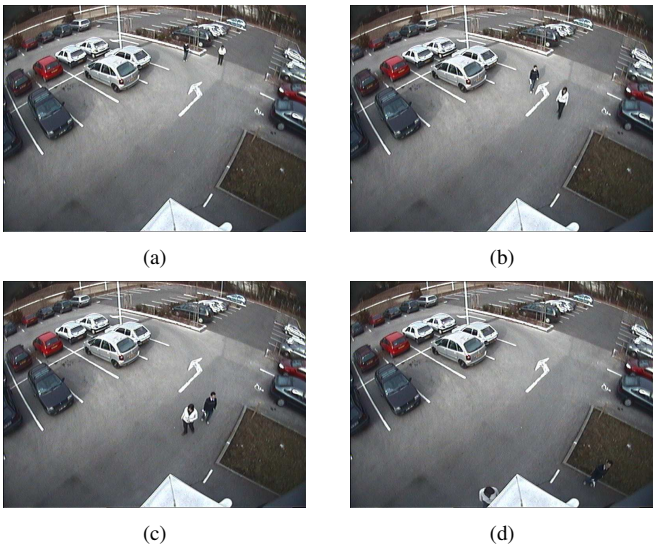
As we are using off-board cameras observing always the same environment, 2 conditions are needed to delete a VRU of the list: it has to be unseen (ie, no observation has been associated to it) for at least the last 3 instants and its estimated position should be outside the intersection.

IV. Experimental results

The methods presented before have been validated on pedestrian tracking on the car park on the Inria lab. In this experiment, two VRUs are walking in the car park in the direction of the camera. They cross and then disappear. Figure 8 shows extracts of the video used for the experiment. Figure 9 shows the result our tracking system using an occupancy grid where object are extracted and tracked. Thanks to the prediction, the trajectories of both pedestrians are correct even if the 2 pedestrians cross. When the pedestrians desappear from the video, their associated target is deleted from the list of VRUs.

V. Conclusion

In this paper, we detail the hardware and software solution we develop to track pedestrians using information about their position given by a set of offboard cameras. The hardware ParkView platform is composed of a set of six off-board analog cameras, installed in a car-park setup such as their field-of-view partially overlap, and three Linux(tm) workstations in



(a) (b) (c) (d)

Fig. 8. Extracts of the video used for the experiments.(a) 2 VRUs start walking in the car park. They have to be tracked (b)(c) The VRUs keep on walking and cross (d), the VRUs are not seen anymore by the camera

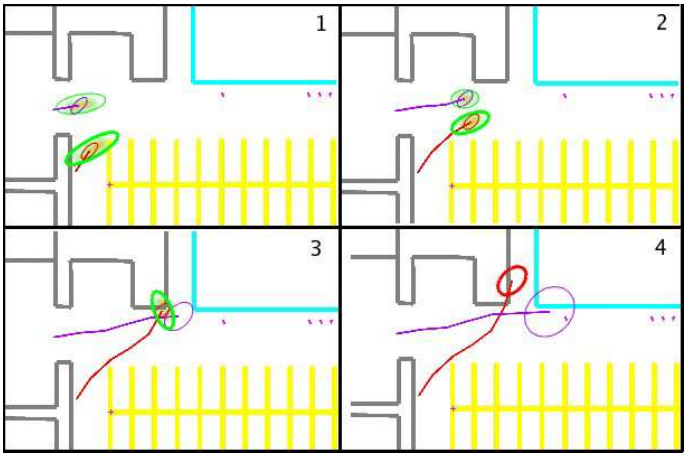


Fig. 9. An example of tracking moving objects over time. Observations (detected objects) are represented by green ellipses. Color lines correspond to estimated tracks. Ellipses in corresponding colors shows incertitude of estimated object positions. (1) tracks initialize with two observations, (2)(3) tracks continue (4), tracks finish, no observation.

charge of the data processing, connected by a standard Local Area Network.

This software solution is divided in 2 main parts:

- A fusion part that allows to use data coming from different sensors in order to compute a better estimation of the position of each pedestrian;
- A tracking part that associates pedestrians currently present in the environment with pedestrians previously present in the environment and estimates the number of pedestrians present in the environment and the position of each pedestrian.

This solution has been implemented and tested and some experimental results have been presented.

The next step will be to extend this software solution in the European Project PReVENT-ProFusion [3]. This software solution will be used as a generic architecture to perform fusion & tracking on demonstrator cars equipped with different types of sensors: lidar, radar

ACKNOWLEDGMENT

This work was partially supported by:

- the french project Predit-PUVAME and Robea-Parknav;
- the european project PReVENT-ProFusion.

Fulgenzi Chiara is supported by a grant from the European Community under the Marie-Curie project VISITOR

REFERENCES

[1] O. Aycard, A. Spalanzani, J. Burlet, T. Fraichard, C. Laugier, D. Raulo, and M. Yguel. Puvame - new french approach for vulnerable road users safety. In *IEEE International Conference on Intelligent Vehicules*, 2006.

[2] C. Blanc, L. Trassoudaine, Y. Le Guilloux, and R. Moreira. Track to track fusion method applied to road obstacle detection. In *International Conference on Information Fusion*, 2004.

[3] H. Cramer, U. Scheunert, and G. Wanielik. Multi sensor fusion for object detection using generalized feature models. In *International Conference on Information Fusion*, 2003.

[4] Alberto Elfes. *Occupancy grids: a probabilistic framework for robot perception and navigation*. PhD thesis, Carnegie Mellon University, 1989.

[5] E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME-Journal of Basic Engineering*, 82:35–45, 1960.

[6] Stan Z. Li. *Markov Random Field Modeling in Image Analysis*. Springer-Verlag, 2001. Series: Computer Science Workbench, 2nd ed., 2001, XIX, 323 p. 99 illus., Softcover ISBN: 4-431-70309-8.

[7] A. Polychronopoulos, U. Scheunert, and F. Tango. Centralized data fusion for obstacle and road borders tracking in a collision warning system. In *International Conference on Information Fusion*, 2004.

[8] S.Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. A K Peters, 1999.

[9] BF. La Scala and A. Farina. Choosing track association method. *Information Fusion*, 3:119–133, 2002.

[10] B. Steux, C. Laurgeau, L. Salesse, and D. Wautier. A vehicule detection and tracking system featuring monocular color vision and radar data fusion. In *IEEE International Conference on Intelligent Vehicules*, 2002.

[11] P. Viola and M. Jones. Fast and robust classification using asymmetric adaboost and a detector cascade. 2002.

[12] M. Yguel, O. Aycard, D. Raulo, and C. Laugier. Grid based fusion of offboard cameras. In *IEEE International Conference on Intelligent Vehicules*, 2006.

[3] http://www.prevent-ip.org/profusion